

A Multi-player Blackjack Game: Pipes

Objectives

- Continue to develop your process programming skills.
- Experiment with client-server interactions.
- Explore pipes and redirection.
- Implement signals and signal handling.

Specification

You are to write a multi-player blackjack program. The program must enforce a maximum of six players. The program must make use of unnamed pipes. The parent process will be the dealer. The children processes will be the players. The game continues until there are no players left. To leave the game a player bets \$0. When a player enters the game, they will tell the dealer the amount of money they will use to begin the game.

The program must implement signal handlers for the termination (SIGINT) and stop (SIGTSTP) signals. If the program receives the termination signal (because all players left the game or from the keyboard), the user should be prompted to shutdown the game or start a new game. If the program receives the stop signal and there are no players left, the program should end gracefully; if there are players still in the game, the signal should be ignored.

The following will be processes of your program:

- Dealer process
 - Shuffle the cards
 - Get initial bets from players
 - Deal the cards to the players and the dealer using a random-number generator
 - Determine which player(s) wins
- Players processes (one child process for each player)
 - Ask the user how much they want to bet
 - Get cards from dealer and tell user
 - Repeat until player stays
 - Ask the user if they want to stay or hit (take a card)
 - Tell the dealer decision (stay or hit)

Usage Clause

blackjack <number_of_players>

Notes

- Do not implement the blackjack concepts of double-down or splitting.
- All players should be able to view the dealer's hand.
- Any non-reentrant function that is interrupted by a signal must be restarted.

The Rules of Blackjack

The following are the rules of blackjack that your program must follow:

- Played with 52 cards, scored as follows:
 - 2-9 → number on card
 - 10, J, Q, K → 10
 - Ace → 1 or 11
 - Score as 11 unless that puts the player over 21, then score as 1
- Each player gets two cards
- Dealer asks each player for user input to determine stay or hit
- Dealer:
 - Always stays on 17 or higher
 - Always hits on 16 or lower
- Betting
 - Minimum bet is \$5
 - Maximum bet is the amount the player has left
 - Bets are in multiples of \$5
- Winning
 - Over 21 → bust (automatically lose)
 - Blackjack → exactly 21 with only first 2 cards dealt pays 3:2 (get your money back plus 1.5 times the amount you bet); if have blackjack, this pays 3:2 even if the dealer also gets 21
 - More than the dealer → win (get your money back plus the amount you bet)
 - Less than the dealer → lose (dealer takes your bet amount)
 - Tie the dealer → draw (you keep your money but win no additional money)

Sample output

Bets in:

Player 1, how much would you like to bet? 5

Player 2, how much would you like to bet? 10

Dealing:

Player 1's hand: [8S][JD]

Player 2's hand: [5H][9C]

Player 1's turn:

Player 1's hand: [8S][10D]

Would you like stay or hit? (S or H): S

Player 1 chooses to stay

Player 2's turn:

Player 2's hand: [5H][9C]

Would you like stay or hit? (S or H): H

Player 2 now has: [5H][9C][10C]

Player 2 busts

Dealer's turn:

Dealer's current hand: [5S][3C]

Dealer chooses to draw

Dealer's current hand: [5S][3C][KC]

Dealer chooses to stay

The Dealer has 18

Player 1 draws (\$0, now has \$100)

Player 2 busted and loses (lost \$10, now has \$90)

<play continues as long as there are still players>

<player 2 enters a bet of \$0>

Player 2 has left the game with \$35